

# STATS 531 project report

## Abstract

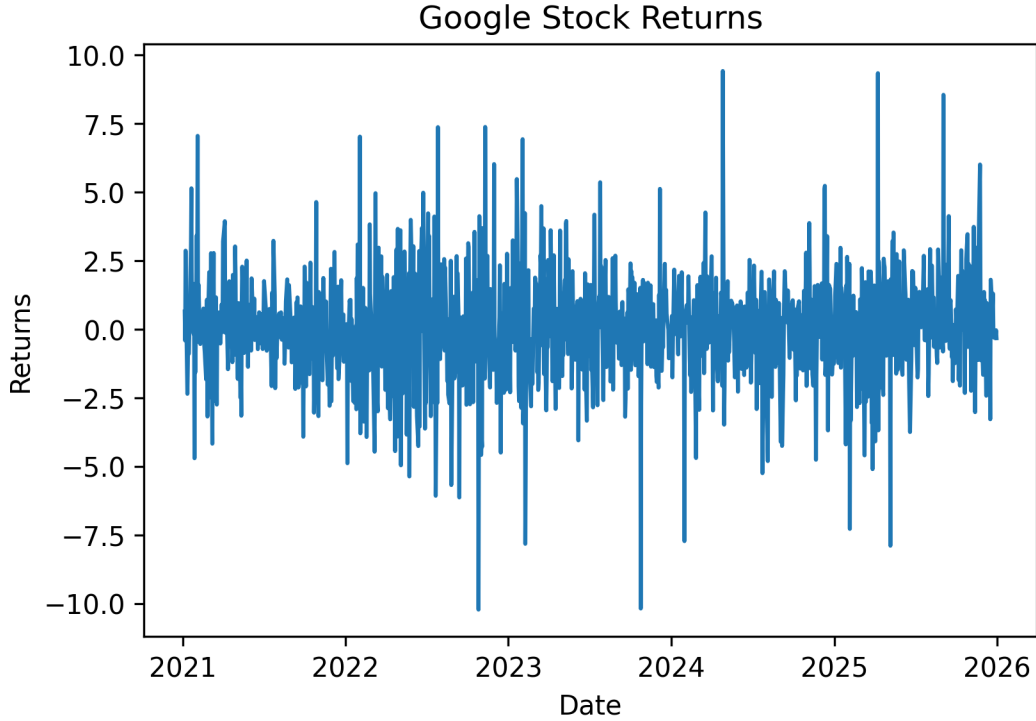
In this report, we explore several ways in which the Bretó stochastic volatility model may be improved, including through using different error distributions and adding different terms to the log volatility. We compare such models to GARCH type models. In doing so, we not only obtain a better model fit compared to GARCH models but also demonstrate the flexibility of the POMP framework by showing how such models can easily be modified and estimated in Pypomp.

## 1 Introduction

It is of general interest in finance to be able to model stock return behavior through stochastic volatility, that is, by modeling the variance of said returns conditional on previous information. This approach is often helpful for stock returns due to volatility clustering, in which the returns series undergoes distinct periods of high or low variance. Thus, by having a model appropriately incorporating stochastic volatility, a practitioner would be better able to assess when a particular asset is more or less risky. To this end, this report will explore variations of the Bretó model and provide a comparison with the GARCH model, which are two well-known volatility models. We ask if there are ways in which the usual Bretó model (Bretó 2014) can be significantly improved upon with slight variation.

## 2 Data

The dataset we will analyze for our investigation will be the Google returns series from January 5, 2021 to December 31, 2025, provided by Yahoo Finance (Yahoo Finance 2026). To access this data, we used `yfinance`, the unofficial API (Aroussi 2017). We use the adjusted close price  $x_n$  to obtain our series of interest, the log returns (which we refer to simply as returns in this report), given by  $y_n = \log(x_n) - \log(x_{n-1})$ . We then demean the data and scale this data by a factor of 100 as preliminary experiments showed worse numerical behavior without scaling. We also note that scaling places data on the same scale as the data in (Ionides 2026a). The stock price and returns series are both shown below.



### 3 GARCH and GJR-GARCH Benchmark Models

Our benchmark model family is the GARCH family of models. For mean zero data as in our case, the  $GARCH(p, q)$  model models the data as

$$Y_n = \varepsilon_n \sqrt{V_n}; \quad V_n = \alpha_0 + \sum_{j=1}^p \alpha_j Y_{n-j}^2 + \sum_{k=1}^q \beta_k V_{n-k}.$$

for a white noise process  $\varepsilon_{1:N}$  (Ionides 2026b). Tables of AIC values for various combinations of  $p, q$  and different error distributions are shown in the supplementary material. To ensure the AICs are comparable, we adjust the `hold_back` parameter to ensure the same data is being utilized to fit each model (see (arch developers 2024)). We find that the best model is the GARCH(1, 1) using the skew-t error distribution, achieving an AIC of 5035.43.

Another useful model is the GJR-GARCH(p, q) model, which is given by

$$V_n = \alpha_0 + \sum_{j=1}^p \alpha_j Y_{n-j}^2 + \sum_{k=1}^q \beta_k V_{n-k} + \gamma \varepsilon_{n-1}^2 \mathbf{1}_{\varepsilon_{n-1} < 0}.$$

This models the idea that negative shocks tend to increase volatility more than positive ones. The GJR-GARCH(0, 1) with skew-t errors provides the overall best benchmark model of the ones tested, obtaining an AIC of 5029.20.

## 4 Basic Bretó Model

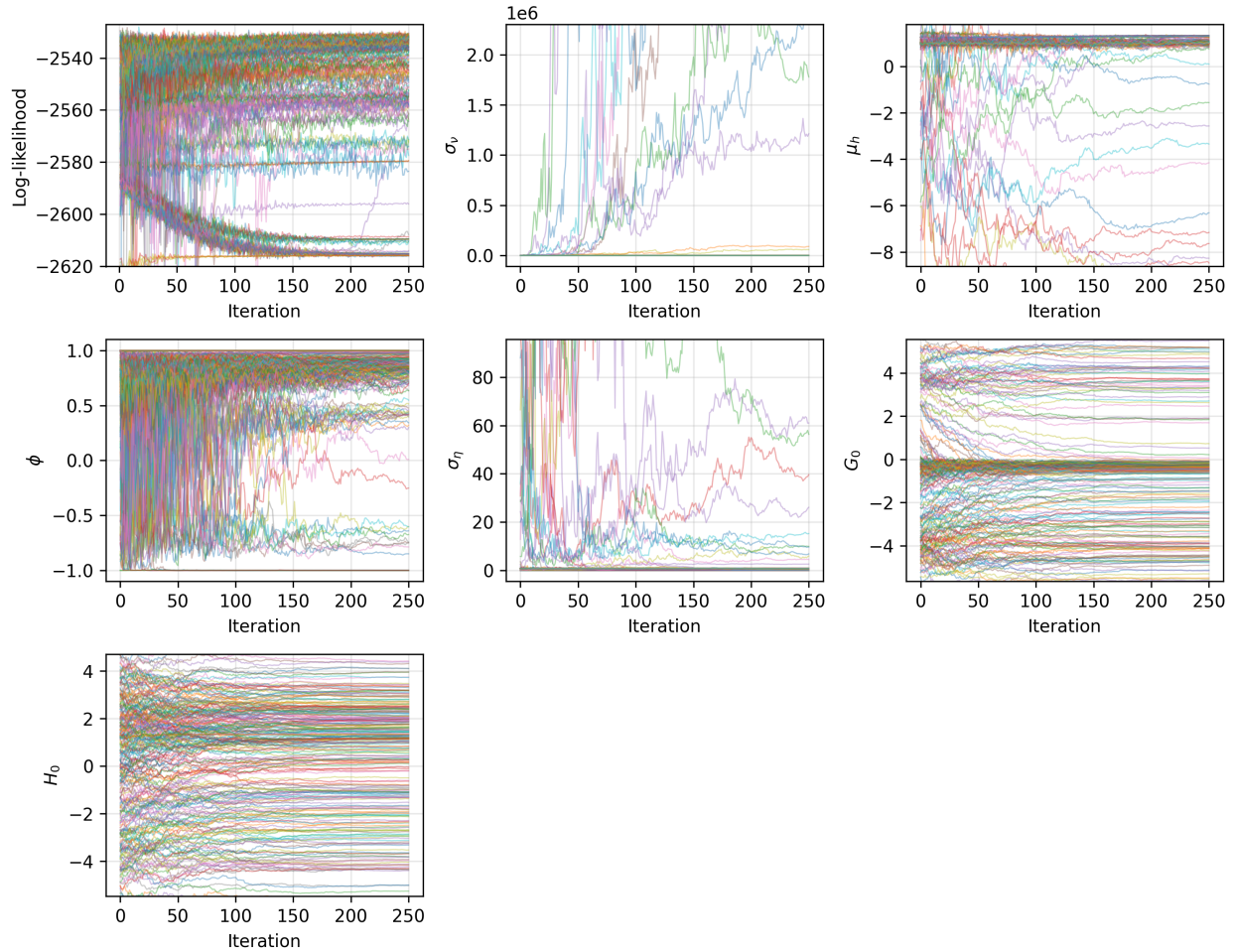
The Bretó model utilizes the concept of leverage, notated as  $R_n$  and defined as the correlation between the return on day  $n - 1$  and the increase in log volatility from day  $n - 1$  to day  $n$ . The model is given by:

$$Y_n = \exp(H_n/2)\varepsilon_n; \quad H_n = \mu_h(1-\phi) + \phi H_{n-1} + \beta_{n-1} R_n \exp(-H_{n-1}/2) + \omega_n; \quad G_n = G_{n-1} + \nu_n$$

where  $\beta_n = Y_n \sigma_\eta \sqrt{1 - \phi^2}$ ,  $\{\varepsilon_n\}$  is iid  $N(0, 1)$  sequence,  $\{\nu_n\}$  is an iid  $N(0, \sigma_\nu^2)$  sequence, and  $\omega_n \sim N(0, \sigma_\eta^2(1 - \phi^2)(1 - R_n^2))$ .

We can use the iterated filtering algorithm to find an estimate for the MLE. Since this model has been commonly utilized in other STATS 531 projects, we rather quickly discuss some of its results to showcase a baseline upon which we can compare subsequent modifications. Local searches are shown in the Supplementary Material. Based off the local search, we build a box of plausible values for the parameters to start the algorithm.

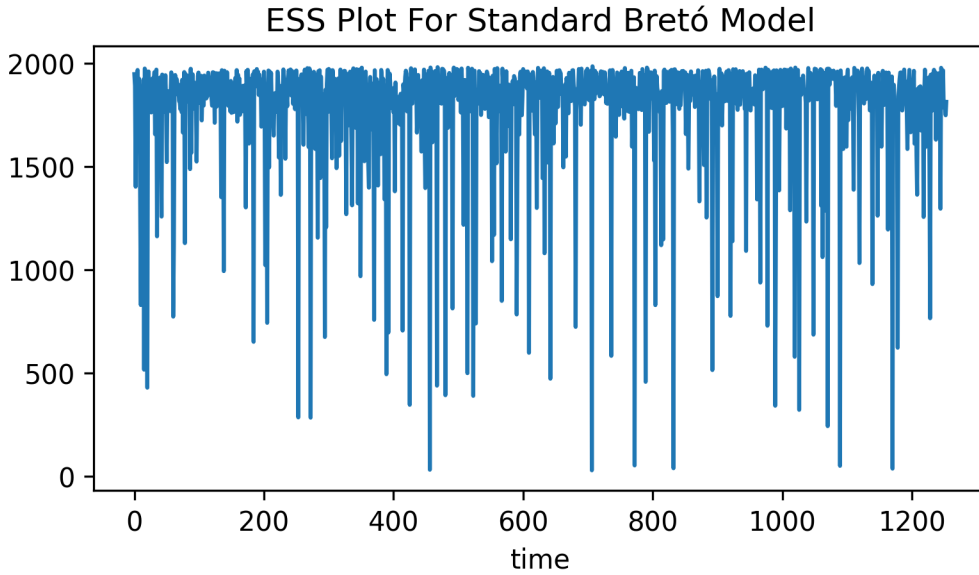
Trace plots from a global search are shown below.



Log likelihoods tend to converge around the -2550 mark though there exist several bad paths, perhaps in part due to a somewhat large search box that gave consideration to parameters resulting in degenerate runs. The algorithm generally favors  $\phi$  somewhere between about 0.80 and 1.00, though small clusters of paths exist elsewhere.  $\sigma_\nu$  and  $\sigma_\eta$  are generally favored to be small, though a few runaway paths favor those parameters to be large.  $\mu_h$  is generally favored to be around 1, and the initial value parameters are generally not very identifiable.

We can also obtain the effective sample size (ESS) for the MLE estimate. We plot the mean ESS over the 10 runs of the filtering algorithm using the MLE estimate. We see numerous dramatic drops, many of which nearly reach 0. Such numerous drops suggest that the model is systematically failing to pick up on some aspect of the data.

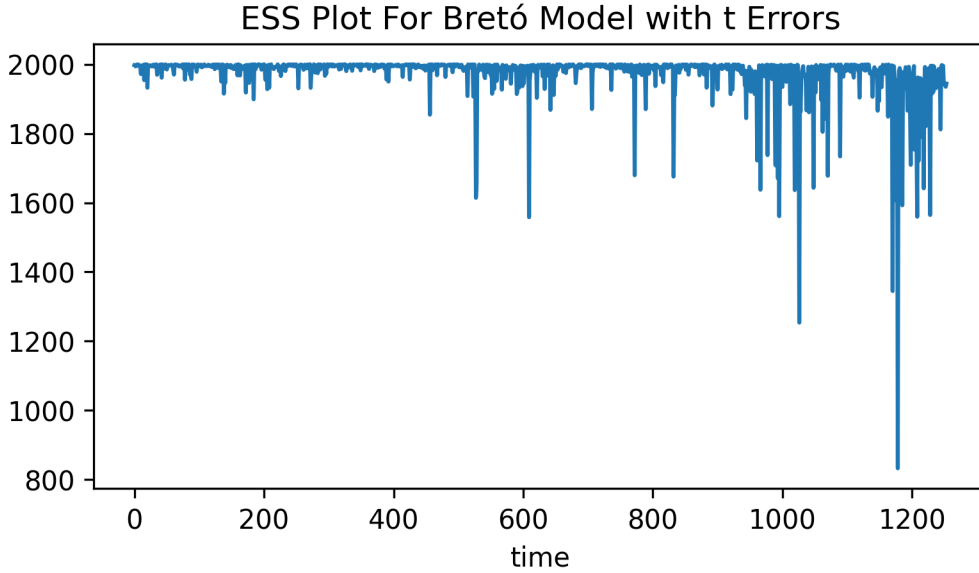
<Figure size 1650x1050 with 0 Axes>



## 5 Bretó Model with t Errors

One potential concern of the previous model is that it assumes  $\varepsilon_n \sim N(0, 1)$  when a heavy-tailed distribution may be more appropriate. In this way, the model may be repeatedly giving too little probability for large innovations, which can result in poor performance on many data points. Thus, we instead assume for the Bretó model  $\varepsilon_n \sim t_\nu$  standardized to have variance 1, where  $\nu$  is the degrees of freedom parameter (unrelated to other parameters in the model). Due to the flexibility of the POMP framework via Pypomp, we can learn  $\nu$  from the data. We then perform a local and global search for the resulting model.

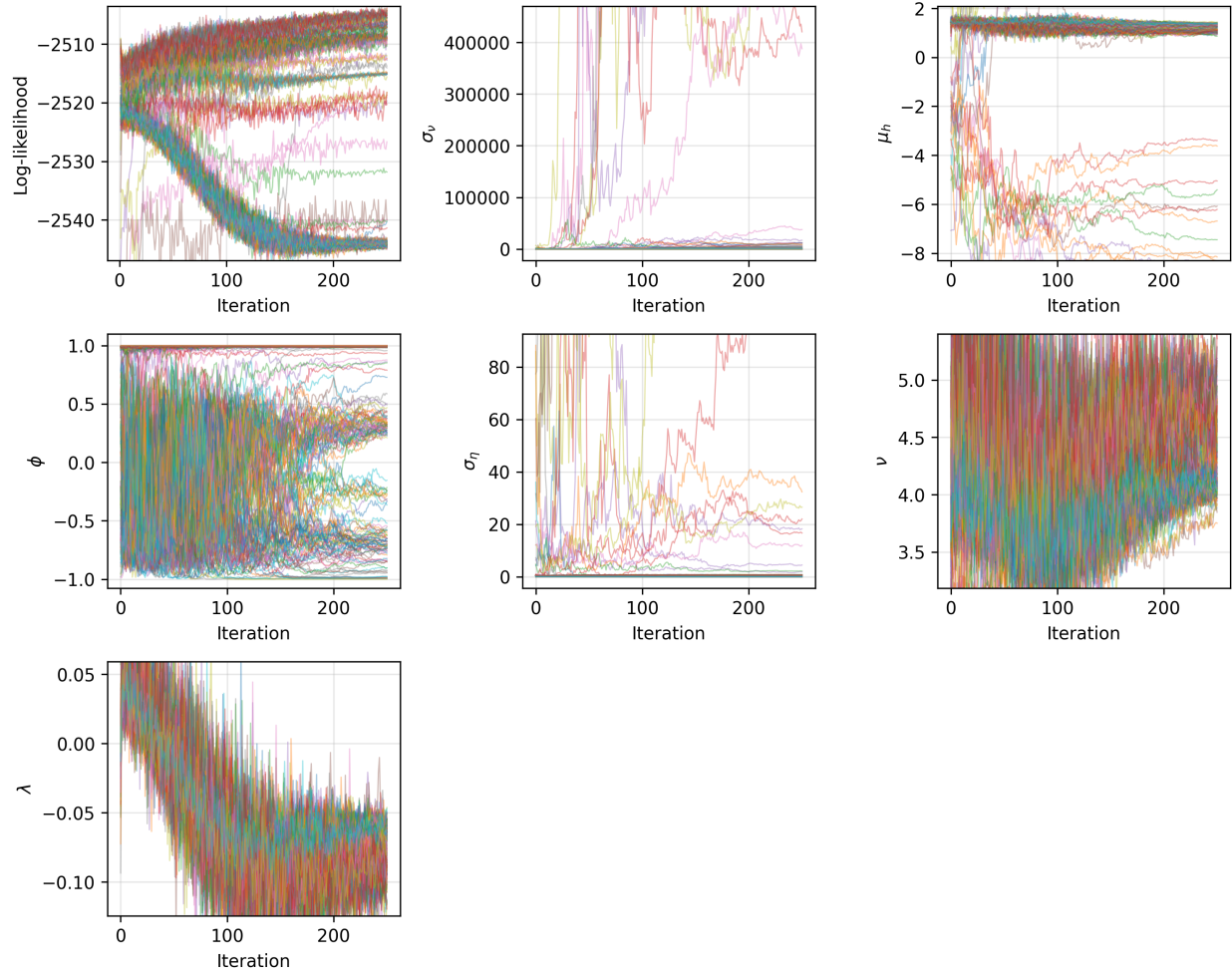
<Figure size 1650x1050 with 0 Axes>



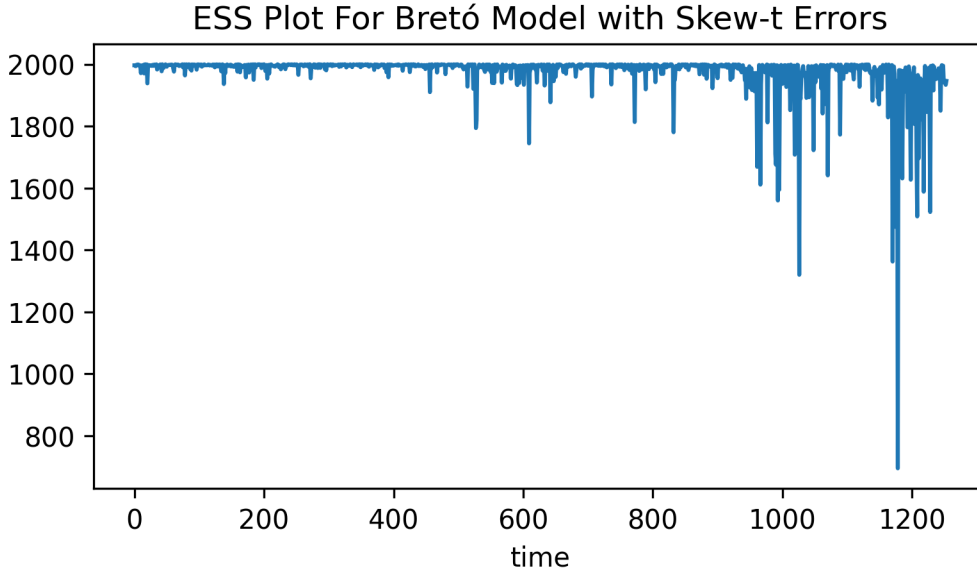
The ESS plot is shown, and the trace plots are found in the Supplementary Material. We see that the ESS vastly outperforms that in the original Bretó model. Furthermore, the MLE achieves a loglikelihood that is over 20 points higher than the previous model. It uses a value  $\nu \approx 4.94$ , which suggests a tail significantly heavier than that of Gaussian. However, the ESS plot seems to get worse for later data, which is concerning.

## 6 Bretó Model with Skew-t Errors

We next investigate whether further improvements can be made in the Bretó model by assuming  $\varepsilon_n$  follow a Skew-t distribution as described in (Hansen 1994). The error distribution in practice need not be symmetric, even if it is known to be mean 0. Some evidence for this is found in (“STATS 531 Final Project - Netflix Returns Analysis” 2025), where the authors found that the skew-t distribution worked well in their GARCH model. The skew-t distribution with parameters  $(\nu, \lambda)$  allows for additional flexibility by introducing a skew parameter  $\lambda$  to skew the  $t_\nu$  distribution. We visualize the global search and the ESS below.



<Figure size 1650x1050 with 0 Axes>



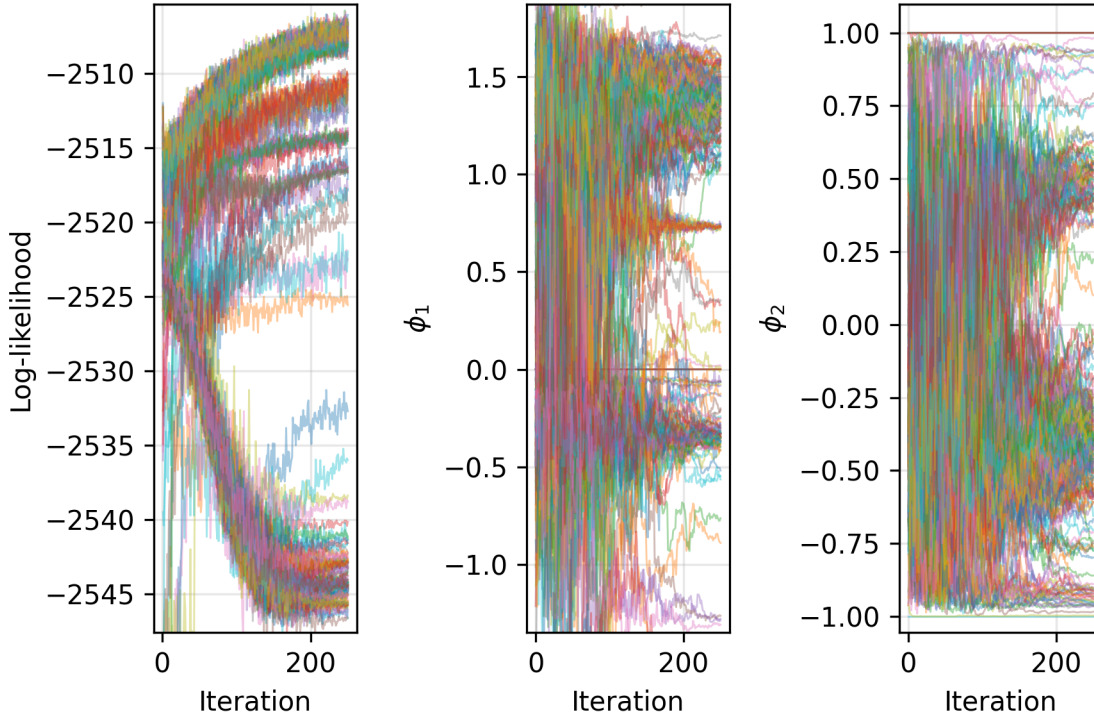
We see that the ESS plot continues to look much better than the standard Breto model, but we can more formally compare how the skew-t error distribution model compares to that with the t distribution by using a likelihood ratio test, noting that the models are nested as the skew-t distribution with  $\lambda = 0$  is the usual t distribution (Hansen 1994). Thus, we use the fact that  $2(\ell_{skew-t} - \ell_t) = 2(-2503.84 + 2506.74) \approx \chi_1^2$  under the null. By doing so, we obtain a p-value of approximately 0.016. We therefore have some evidence to suggest that the skew may be nontrivial, noting that these are only estimates of the log-likelihood, and a more thorough computation may diminish the evidence in favor of the skew-t distribution.

## 7 Bretó Model with AR(2) Log Volatility

We next want to determine if the Bretó model can be modified through a way other than its error distribution and still give a substantial improvement in performance. To do so, we attempt to include an AR(2) term, such that  $H_n$  can be written as

$$H_n = \mu_h(1 - \phi_1 - \phi_2) + \phi_1 H_{n-1} + \phi_2 H_{n-2} + \beta_{n-1} R_n \exp(-H_{n-1}/2) + \omega_n$$

where  $\beta_n = Y_n \sigma_\eta$ , and  $\omega_n \sim N(0, \sigma_\eta^2(1 - R_n^2))$ . To do so, we expand the state variable as  $X_n = (G_n, Y_n, H_n, H_{n-1})$ . To ensure that the AR(2) process in  $H_n$  is stationary, we have to transform the coefficients  $\phi_1, \phi_2$  to stationary region (shown in (Hanck 2015)). We use  $\varepsilon_n \sim t_\nu$  due to its clear success over normal errors.



The log-likelihood obtained for the MLE estimate for this model is approximately  $-2506.16$ , making it hardly better than the model without the AR(2) term. The ESS plot is roughly the same as that of the previous models and is shown in the supplementary material.

## 8 Summary of Results

We summarize our likelihood and AIC results in the table below, where we indeed see that the skew-t error distribution provides the best AIC.

Model	Log-likelihood	Num. Parameters	AIC
Standard Bretó	-2530.82	6	5073.65
t-Errors	-2506.74	7	5027.48
Skew-t Errors	-2503.84	8	5023.69
AR(2)-type	-2506.16	8	5028.32

## 9 Direct Comparison of Modified Bretó Model with GARCH

To more directly compare log-likelihoods and AIC with the GARCH models, we must ensure that we fit models to the same dataset, so we then refitted our most successful model, the skew-t errors model, this time omitting the first 4 points in the dataset, since the GARCH models were fit with `hold_back = 4`. (We did not refit our other models in this way due to time constraints.) We thus

obtain a log-likelihood of about  $-2496.19$  and thus an AIC of about  $2(2496.19 + 8) = 5008.38$ , beating the GARCH(1, 1) and GJR-GARCH(0, 1) models.

## 10 Discussion and Limitations

We have seen several ways in which the Bretó model can be improved relatively easily using the Pypomp implementation, which allows for experimentation with additional terms or different error distributions. We see that significant gain can come from adjusting the error distribution especially. The t-distribution is a large improvement over normal errors, and the skew-t distribution may even further improve upon that, though results in this regard may be mixed, perhaps depending on the setting. Here, the skew parameter was somewhat small (approximately  $-0.08$ ).

An additional model was attempted in the Supplementary Material, where we include an asymmetry term that increases  $H_n$  only when  $Y_{n-1} < 0$ , similar to that found in the GJR-GARCH model, though it was unhelpful in this case.

Our analysis has some limitations. The ESS plot for each model shows decrease in performance near the end of the timeframe. The persistence of this despite various modifications to the Bretó model may suggest that a more foundational issue is at play and an entirely different model may resolve this.

## 11 Connection to Previous Projects

There have been several STATS 531 projects that have previously compared the Bretó model with GARCH-type models. We used the idea of (“Volatility Analysis on Bitcoin Returns: A Fear & Greed Index Perspective” 2025) to use the t distribution for errors instead of the normal distribution, which inspired us to try using different error distributions for our model. (“Volatility Analysis on Bitcoin Returns: A Fear & Greed Index Perspective” 2025) also experimented with an additional term in the  $H_n$  expression, which we took in another direction by adding different terms such as lag terms. (“STATS 531 Final Project - Netflix Returns Analysis” 2025) also used the skew-t distribution to improve model fit, but for the GARCH model. We took it a step further by implementing a JAX-friendly skew-t distribution to use with Pypomp and showed it may be helpful for models such as the Bretó model. Other projects using financial data almost exclusively, to our knowledge, compared standard existing financial models without much modification.

## 12 Supplemental Material

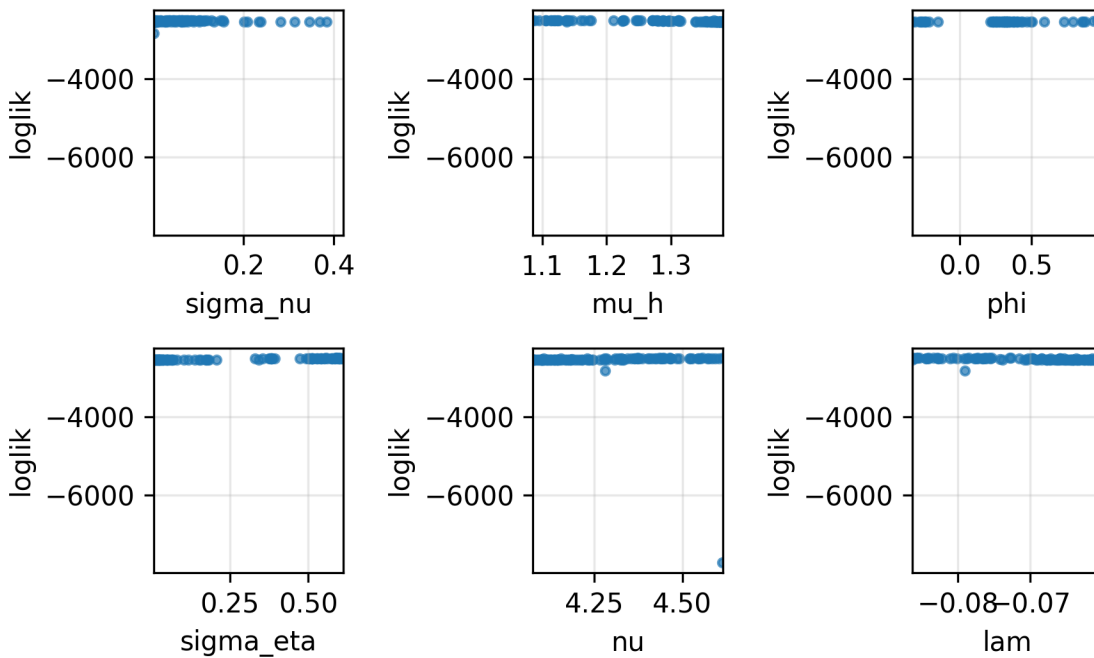
	GARCH0	GARCH1	GARCH2	GARCH3	GARCH4
ARCH0	NaN	NaN	NaN	NaN	NaN
ARCH1	5075.160886	5037.238565	5038.641381	5040.641381	5042.641382
ARCH2	5073.229297	5039.238565	5040.366924	5042.366924	5044.366924
ARCH3	5064.088307	5041.238565	5042.351094	5044.297417	5043.098102
ARCH4	5056.176693	5043.238565	5044.351094	5046.297417	5044.191414

	GARCH0	GARCH1	GARCH2	GARCH3	GARCH4
ARCH0	NaN	NaN	NaN	NaN	NaN
ARCH1	5074.966833	5035.431720	5036.994172	5038.994172	5040.994172
ARCH2	5072.779658	5037.431721	5038.809176	5040.809176	5042.809177
ARCH3	5062.530474	5039.431659	5040.752774	5042.502769	5040.982237
ARCH4	5054.927338	5041.431659	5042.752774	5044.502768	5042.151663

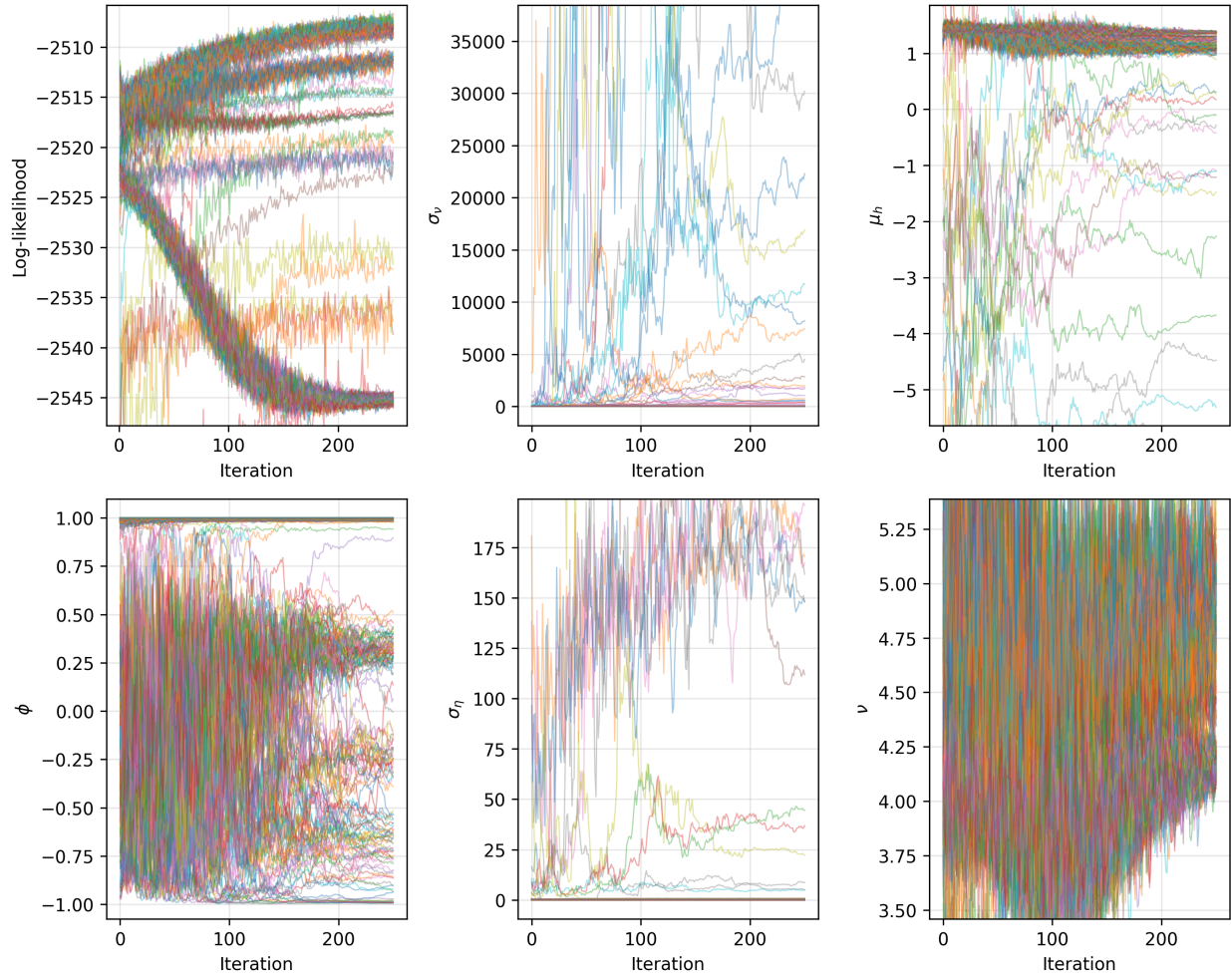
	GARCH0	GARCH1	GARCH2	GARCH3	GARCH4
ARCH0	5078.579986	5029.204941	5031.372044	5033.372044	5035.372045
ARCH1	5076.961375	5029.766632	5031.767249	5033.767249	5035.767250
ARCH2	5074.746421	5031.766632	5033.693419	5035.693419	5037.693419
ARCH3	5064.362006	5033.766632	5035.693419	5037.693419	5039.693419
ARCH4	5056.789460	5035.766632	5037.693419	5039.693419	5041.693420

## 12.1 Global Search Visualization for the Skew-t Error Model

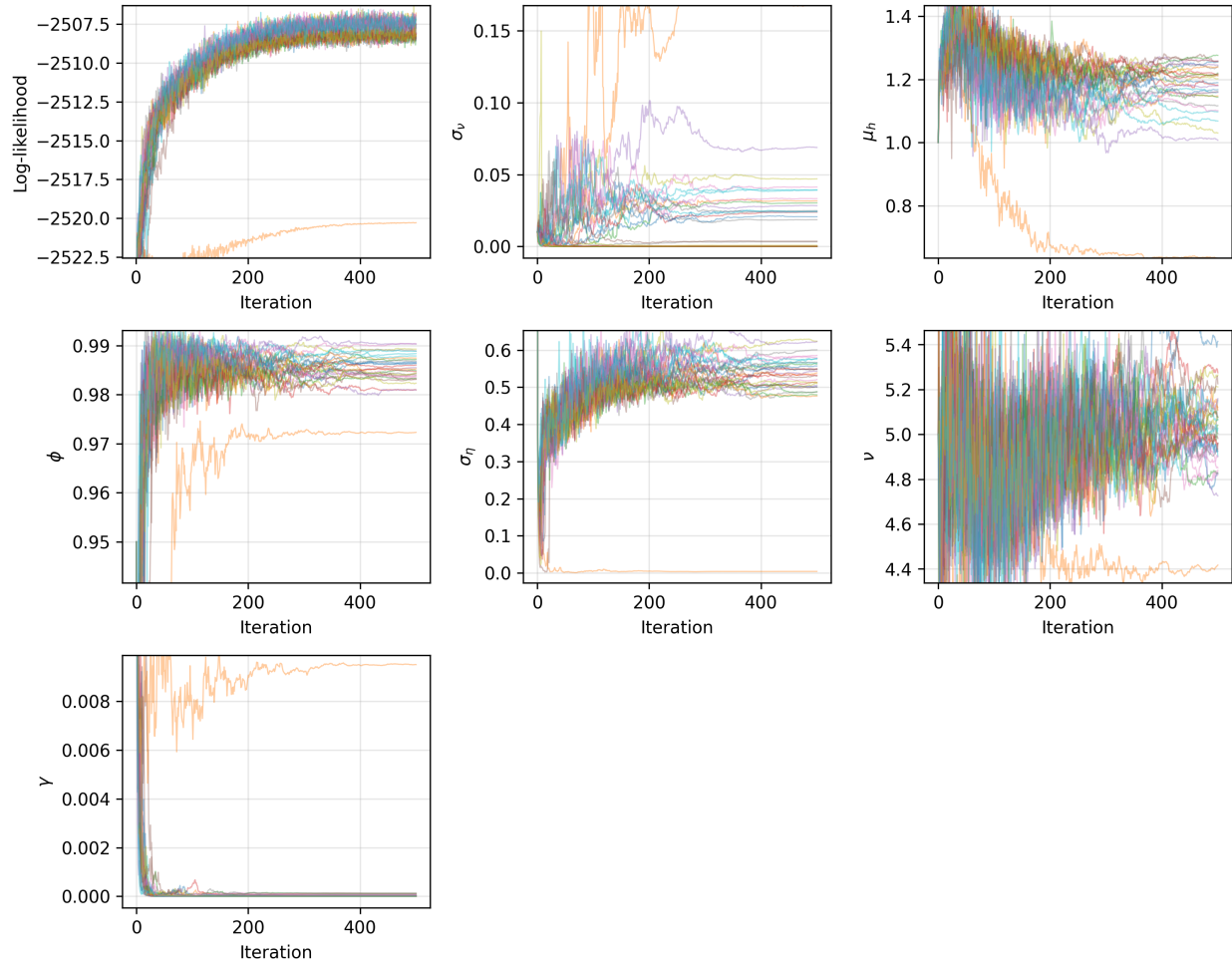
### Global Search Visualization - Skew-t Error Model

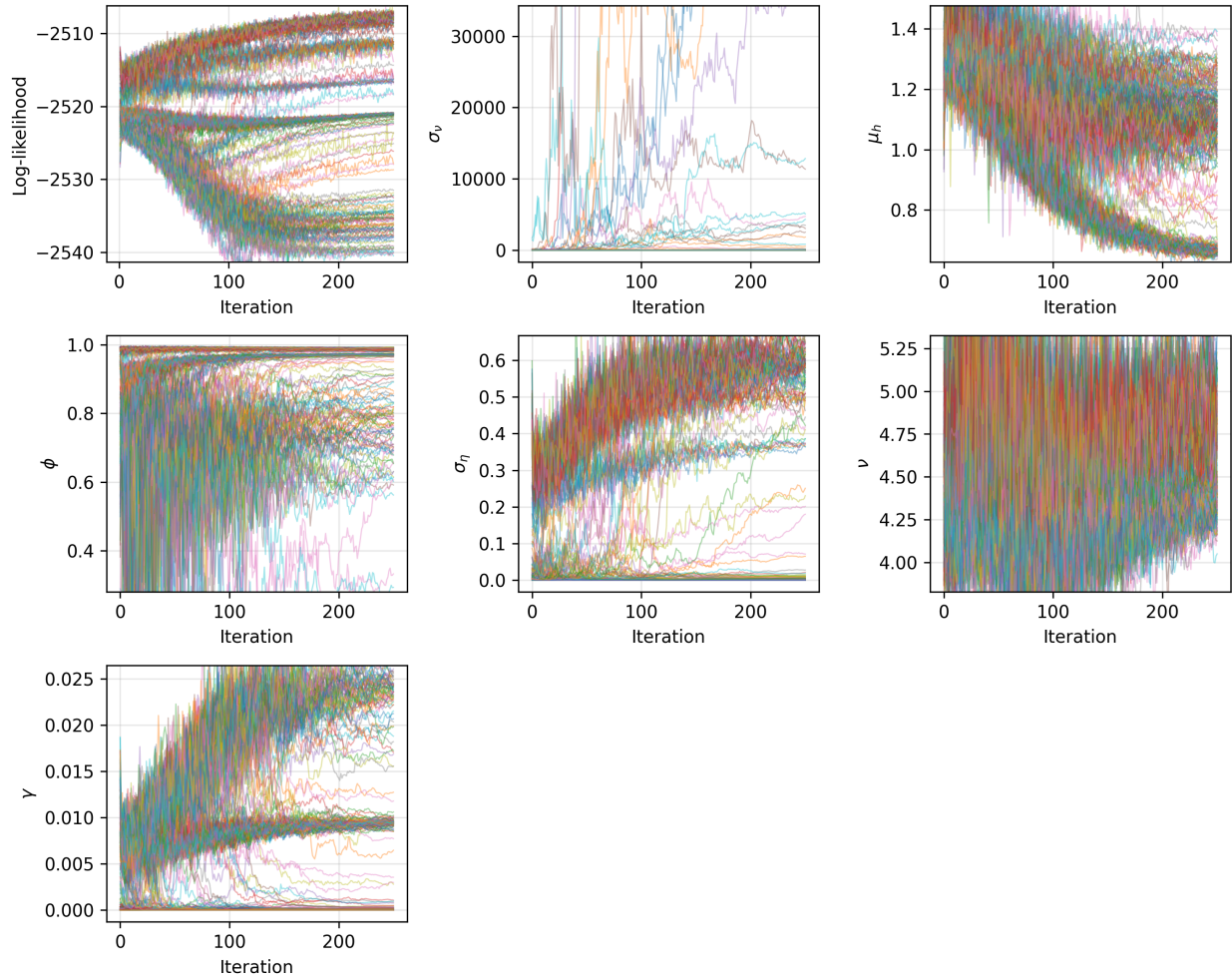


## 12.2 Extra Trace plots



### 12.3 Bretó model with GJR asymmetry term





```
{'index': 148,
 'mle': sigma_nu      0.065336
 mu_h      1.026408
 phi       0.990411
 sigma_eta 0.650377
 G_0      -2.927807
 H_0       1.652570
 nu        4.969345
 gamma     0.000004
 loglik    -2506.635087
 loglik_se 0.115901
 Name: 148, dtype: float64}
```

## 12.4 MLE results

The standard Bretó model:

```
{'index': 73,  
  'mle': sigma_nu      0.000079  
mu_h      1.009749  
phi       0.862305  
sigma_eta 0.809751  
G_0      -0.206261  
H_0      0.833257  
loglik    -2530.824348  
loglik_se  0.131497  
Name: 73, dtype: float64}
```

Bretó model with t-errors:

```
{'index': 69,  
  'mle': sigma_nu      0.075518  
mu_h      1.045580  
phi       0.990300  
sigma_eta 0.642378  
G_0      -2.808092  
H_0      1.579443  
nu        4.940762  
loglik    -2506.740027  
loglik_se  0.064587  
Name: 69, dtype: float64}
```

Bretó model with skew-t errors:

```
{'index': 15,  
  'mle': sigma_nu      0.069035  
mu_h      1.020328  
phi       0.990777  
sigma_eta 0.673780  
G_0      -3.136283  
H_0      1.750626  
nu        4.655053  
lam       -0.086185  
loglik    -2503.842984  
loglik_se  0.114103  
Name: 15, dtype: float64}
```

The AR(2) style volatility model:

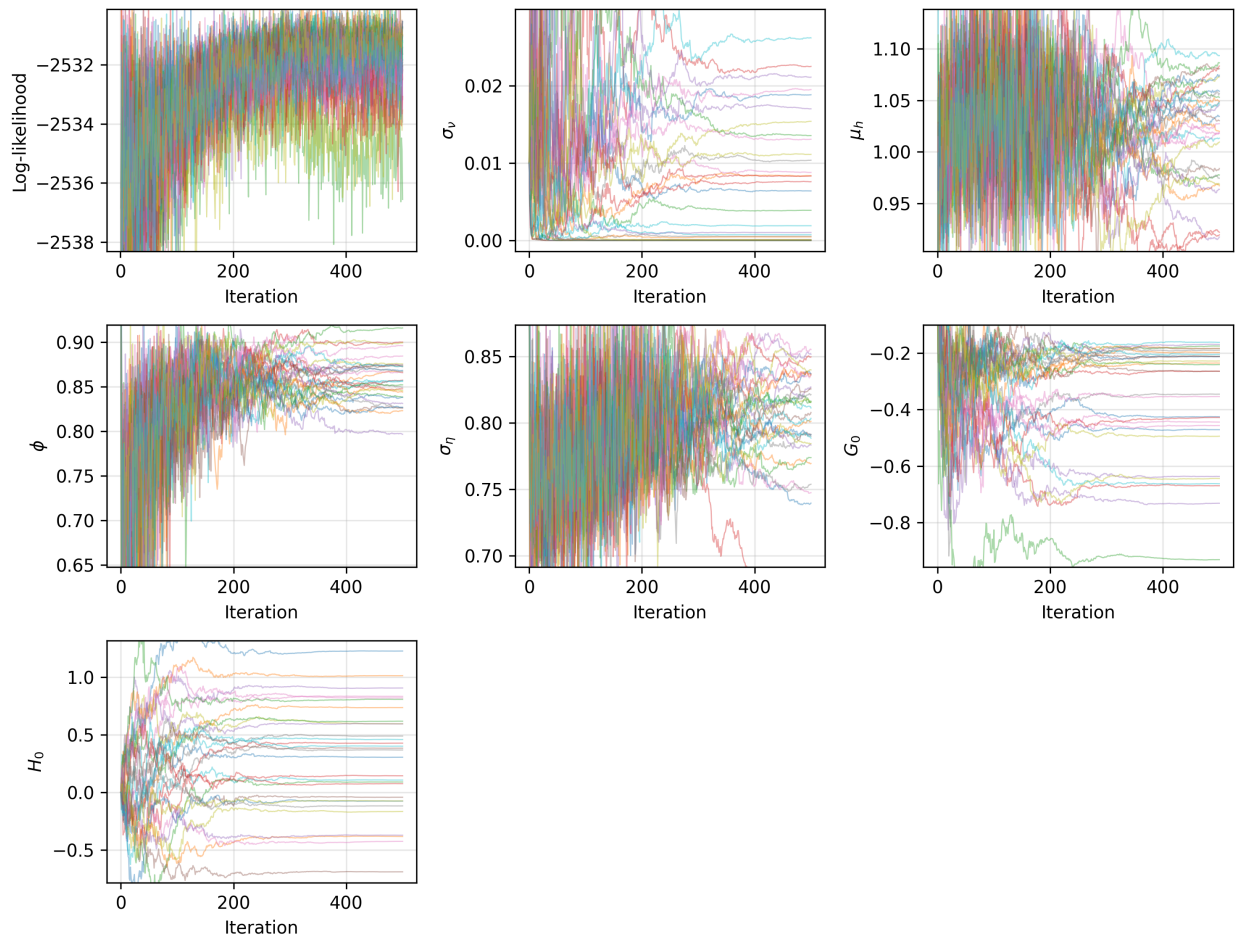
```

{'index': 14,
 'mle': sigma_nu      0.087668
 mu_h      1.072775
 phi_1     1.410520
 phi_2     -0.416343
 sigma_eta 0.052365
 G_0       -2.950950
 H_0       1.560690
 nu        5.377692
 loglik    -2506.159959
 loglik_se 0.133876
 Name: 14, dtype: float64}

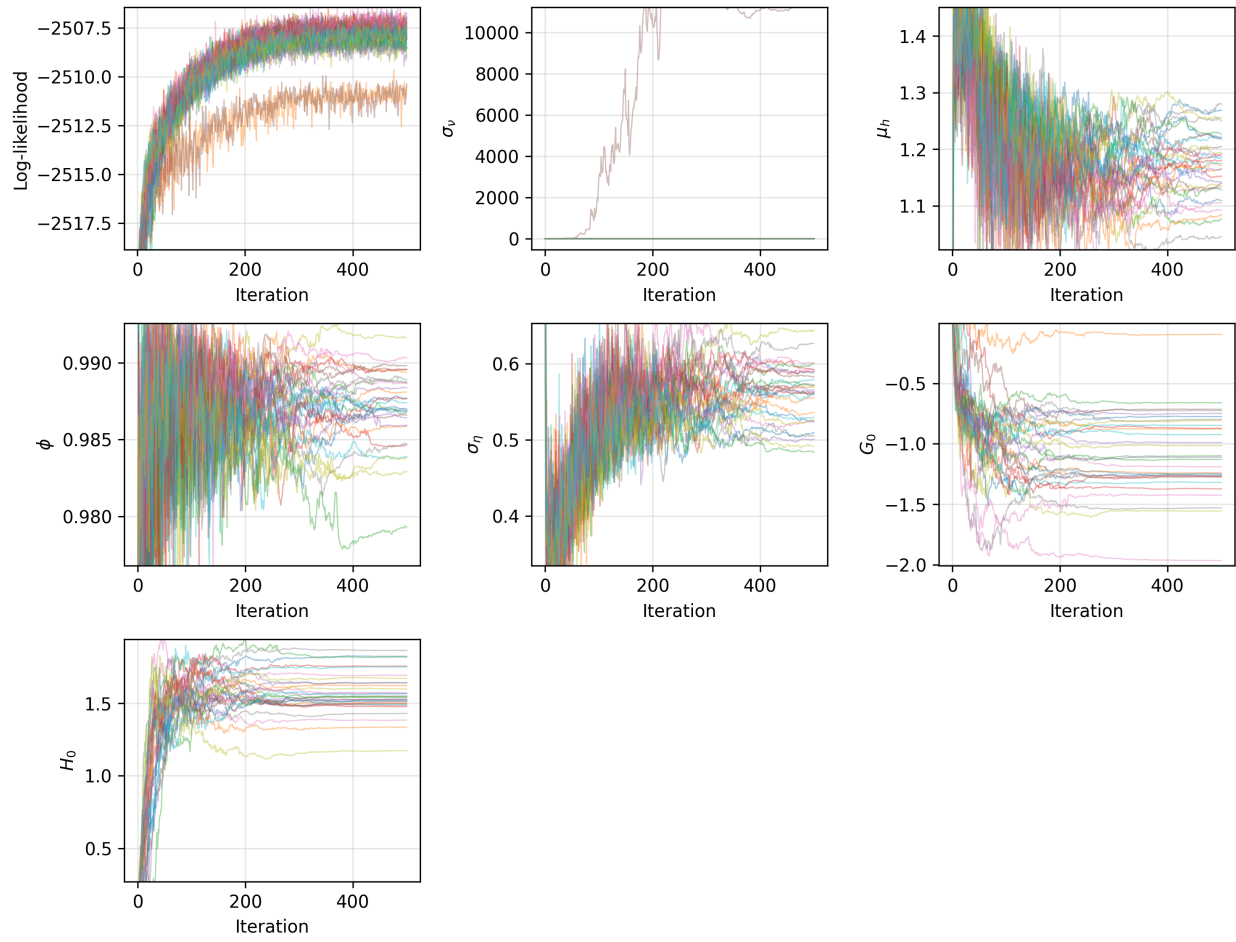
```

## 12.5 Local Searches

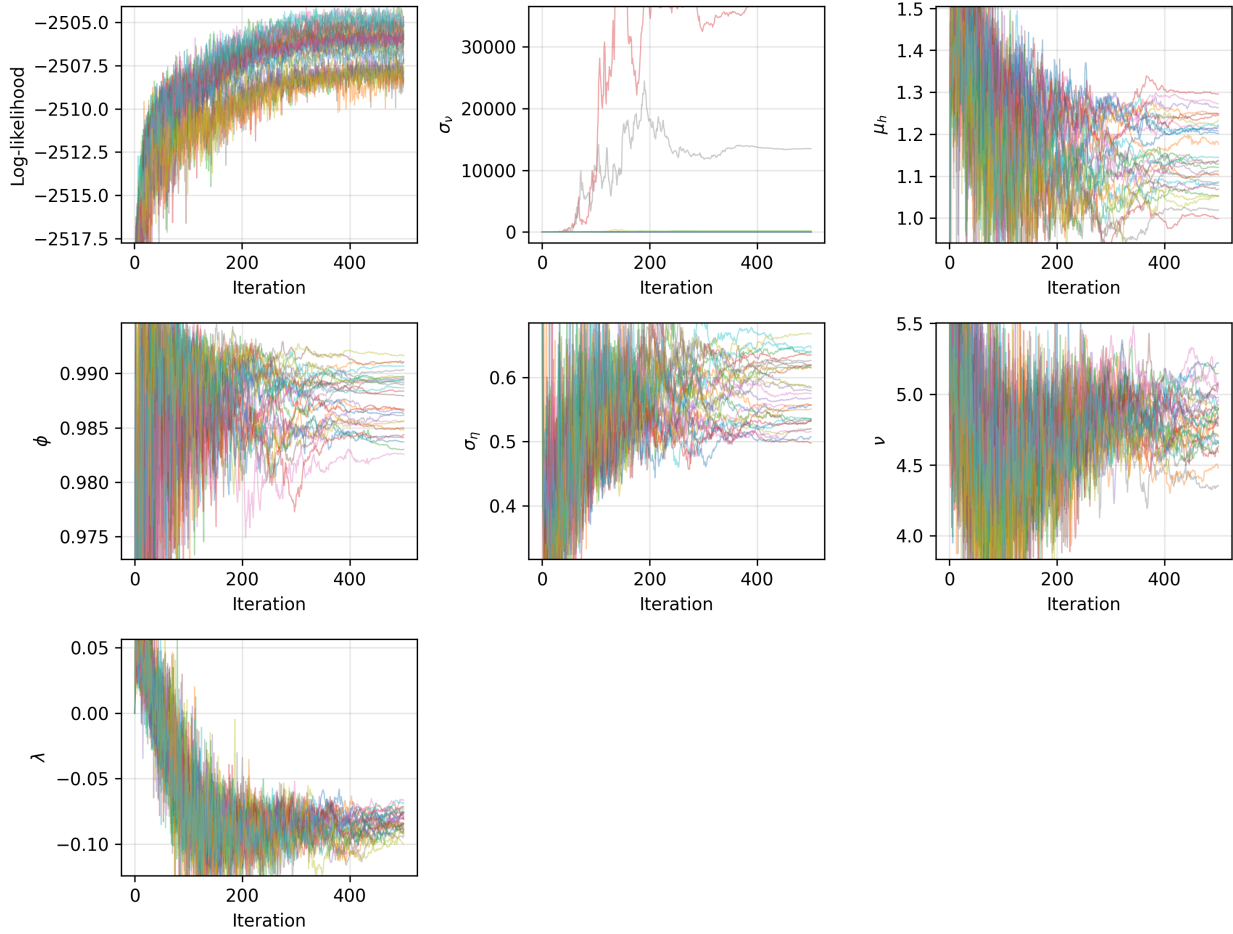
Standard Breto Model Local Search Results



### Breto Model With t-Errors Local Search Results



### Breto Model With Skew-t Errors Local Search Results



## 13 References

ChatGPT was used for coding assistance, specifically with proofreading model implementations and global and local search workflows to check for consistency and correctness, implementing the JAX-compatible skew-t distribution, various plotting and document formatting help, for code edits to improve numerical robustness (e.g. preventing division by zero due to floating point arithmetic), and for the setting up the `from_est` and `to_est` functions for the AR(2)-type model.

arch developers. 2024. *Arch Documentation: Univariate Volatility Models*. Read the Docs. <https://arch.readthedocs.io/en/latest/univariate/introduction.html>.

Aroussi, Ran. 2017. “Yfinance: Yahoo Finance Market Data Downloader.” <https://github.com/ranaroussi/yfinance>.

Bretó, Carles. 2014. “On Idiosyncratic Stochasticity of Financial Leverage Effects.” *Statistics & Probability Letters* 91: 20–26.

Hanck, Christoph. 2015. “A Proof for the Stationarity of an AR(2).” <https://stats.stackexchange.com/questions/118019/a-proof-for-the-stationarity-of-an-ar2>.

- Hansen, Bruce E. 1994. "Autoregressive Conditional Density Estimation." *International Economic Review* 35 (3): 705–30. [https://users.ssc.wisc.edu/~bhansen/papers/ier\\_94.pdf](https://users.ssc.wisc.edu/~bhansen/papers/ier_94.pdf).
- Ionides, Edward L. 2026a. "Course Notes: Source File (Lecture 17)." <https://github.com/ionides/531w26/blob/main/17/source.qmd>.
- . 2026b. "Lecture 17 Slides." <https://github.com/ionides/531w26/blob/main/17/slides.pdf>.
- OpenAI. 2026. "ChatGPT (GPT-5.3)." <https://chat.openai.com>.
- "STATS 531 Final Project - Netflix Returns Analysis." 2025. [https://ionides.github.io/531w25/final\\_project/project08/blinded.html](https://ionides.github.io/531w25/final_project/project08/blinded.html).
- "Volatility Analysis on Bitcoin Returns: A Fear & Greed Index Perspective." 2025. [https://ionides.github.io/531w25/final\\_project/project15/blinded.html#26\\_Simple\\_Stochastic\\_Volatility\\_Model\\_with\\_Student's\\_t-Distribution](https://ionides.github.io/531w25/final_project/project15/blinded.html#26_Simple_Stochastic_Volatility_Model_with_Student's_t-Distribution).
- Yahoo Finance. 2026. "Alphabet Inc. (GOOG) Historical Data." <https://finance.yahoo.com/quote/GOOG/history/>.